FIG. 1

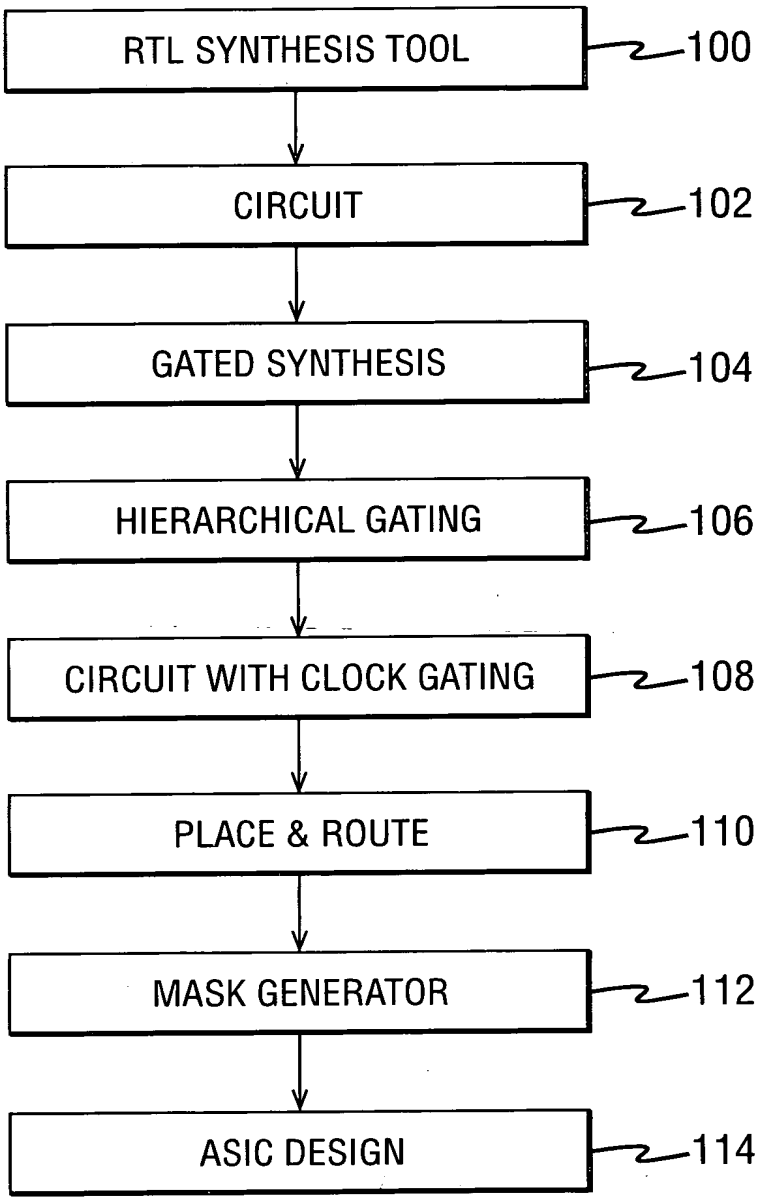
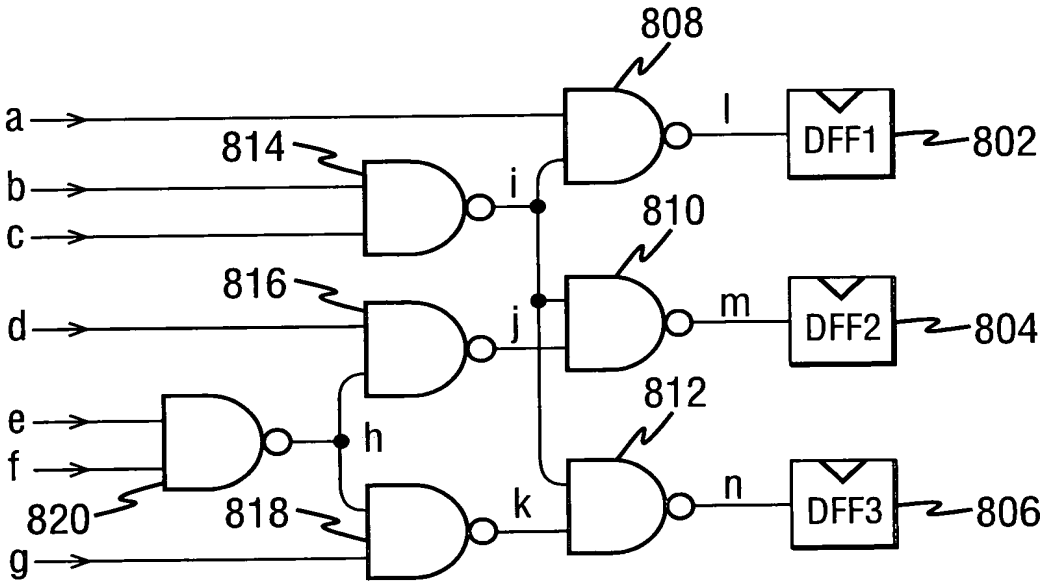


FIG. 2


808
a
814
l
DFF1
802
b
i
c
810
816
m
DFF2
804
d
j
812
e
h
f
n
DFF3
806
820
818
k
g

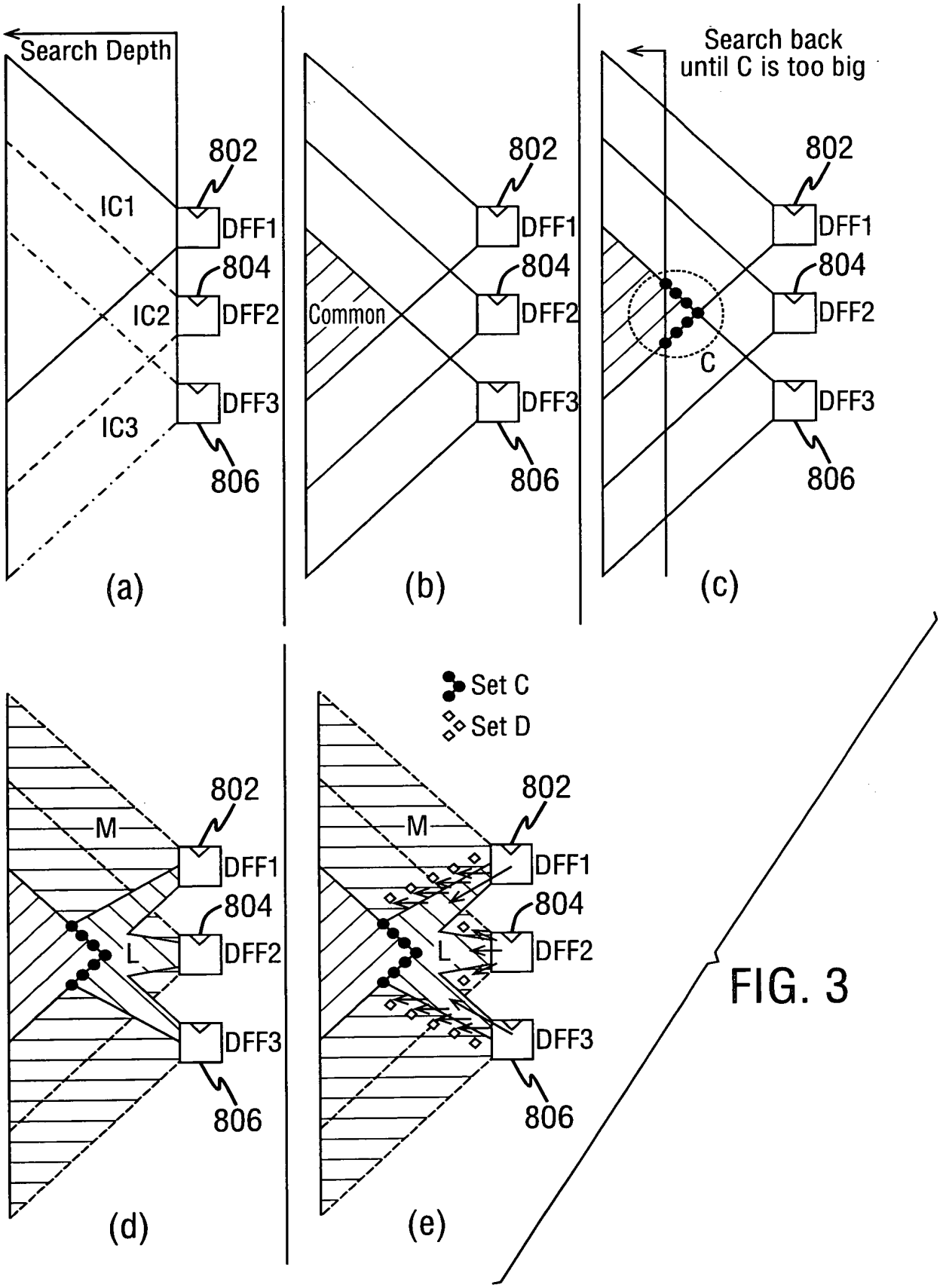

Search Depth
802
IC1
DFF1
804
IC2
DFF2
DFF3
IC3
806
(a)
802
DFF1
804
Common
DFF2
DFF3
806
(b)
Search back
until C is too big
802
DFF1
804
DFF2
C
DFF3
806
(c)
802
M
DFF1
804
L
DFF2
DFF3
806
(d)
Set C
Set D
802
M
DFF1
804
L
DFF2
DFF3
806
(e)

FIG. 3

FIG. 4

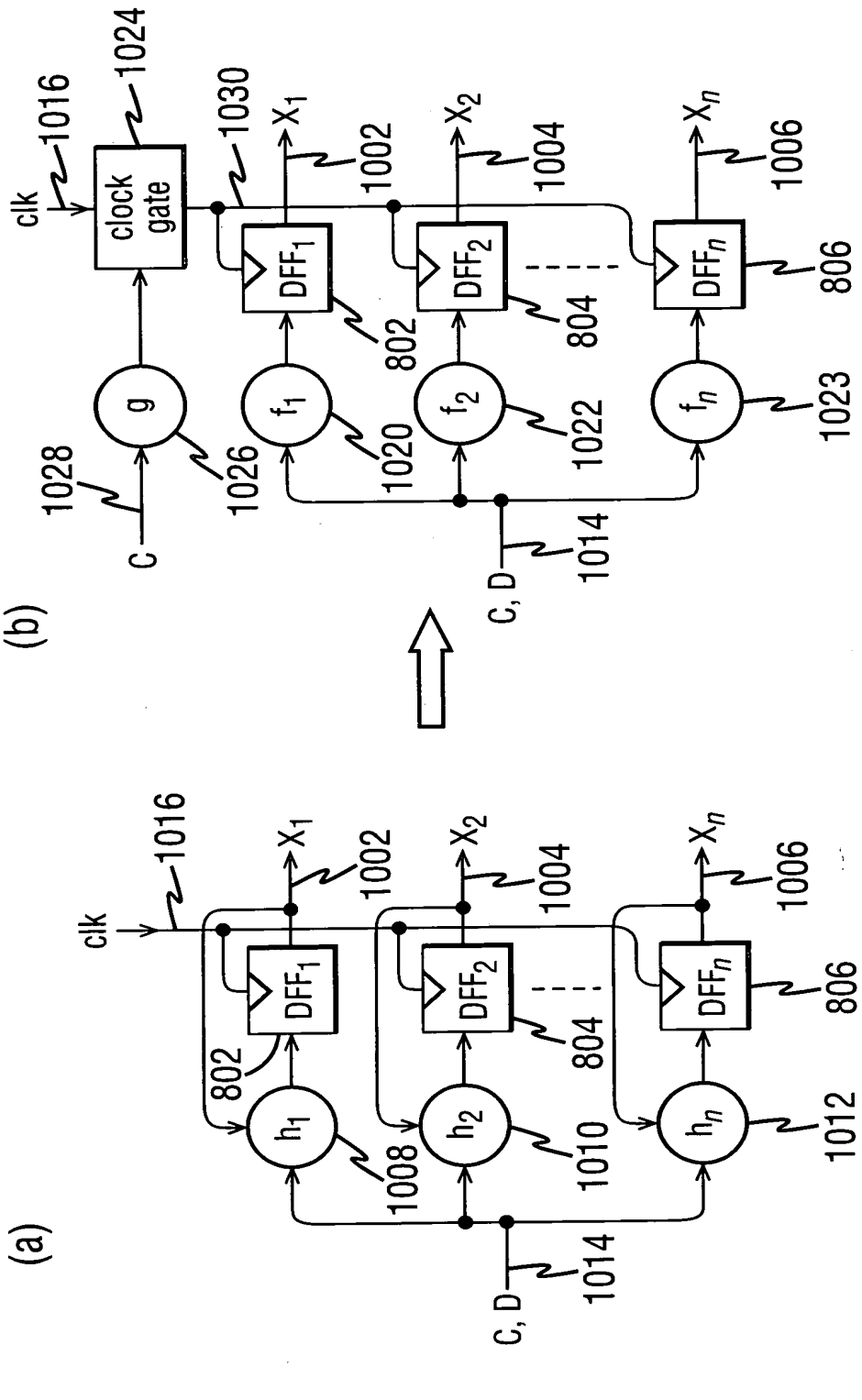

(a)
clk
1016
X_1
1002
X_2
1004
X_n
1006
DFF_1
DFF_2
DFF_n
802
804
806
h_1
h_2
h_n
1008
1010
1012
C, D
1014
(b)
clk
1016
clock gate
1024
1030
g
1028
C
1026
f_1
f_2
f_n
1020
1022
1023

FIG. 5

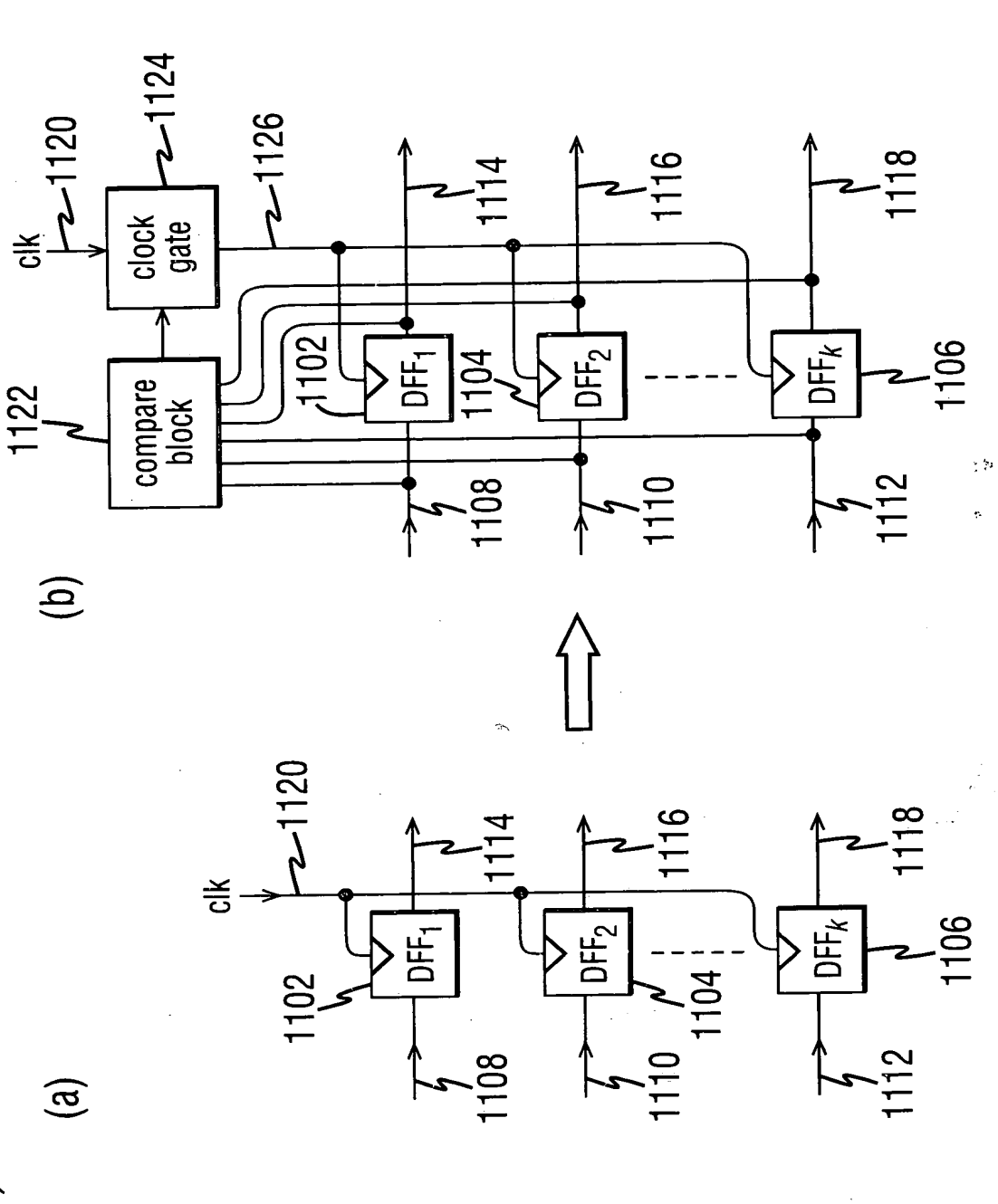

(a)
(b)
clk
1120
clock gate
1124
1126
compare block
1122
1102
DFF1
DFF2
DFFk
1104
1106
1108
1110
1112
1114
1116
1118

FIG. 6

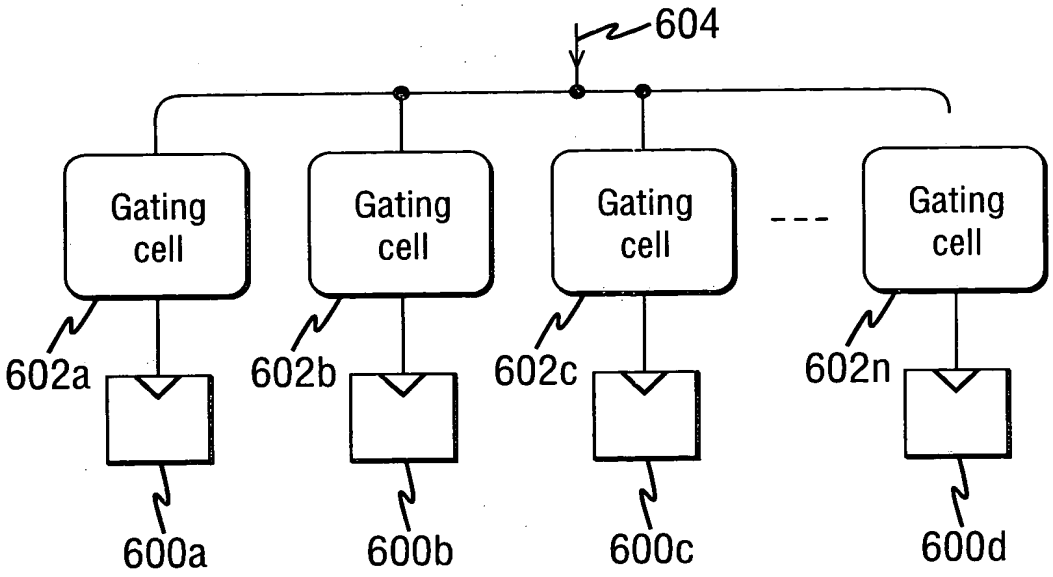

604
Gating cell
Gating cell
Gating cell
Gating cell
602a
602b
602c
602n
600a
600b
600c
600d

FIG. 7

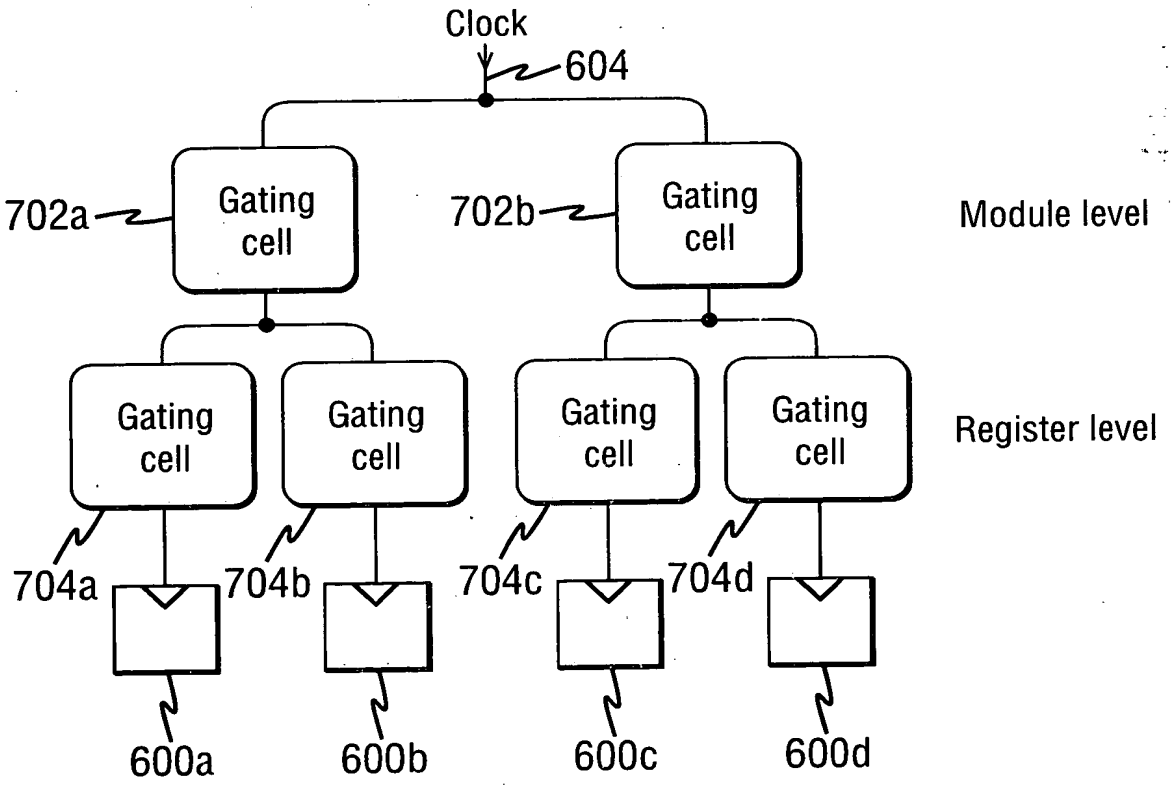

Clock
604
702a
Gating cell
702b
Gating cell
Module level
Gating cell
Gating cell
Gating cell
Gating cell
Register level
704a
704b
704c
704d
600a
600b
600c
600d

FIG. 8

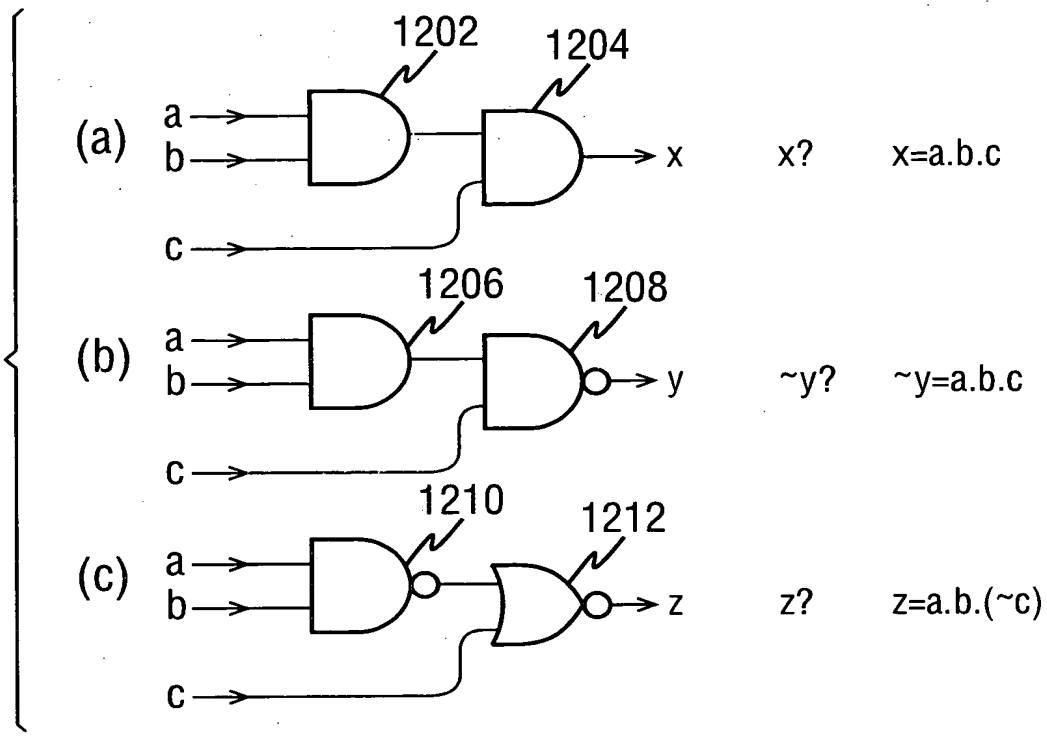

(a)
1202
1204
a
b
c
x
x?
x=a.b.c
(b)
1206
1208
a
b
c
y
~y?
~y=a.b.c
(c)
1210
1212
a
b
c
z
z?
z=a.b.(~c)

FIG. 9

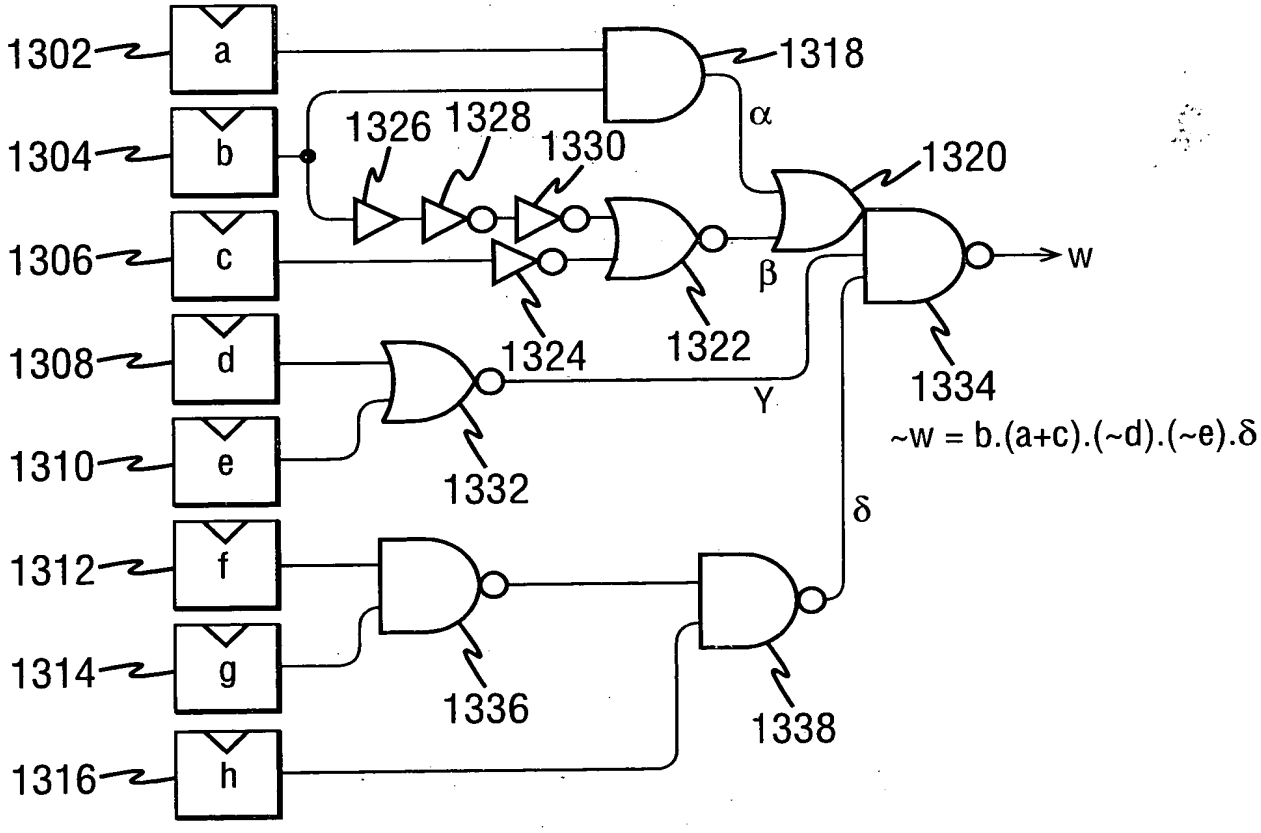

1302
a
1304
b
1306
c
1308
d
1310
e
1312
f
1314
g
1316
h
1318
α
1326
1328
1330
1320
β
w
1324
1322
γ
1334
~w = b.(a+c).(~d).(~e).δ
1332
δ
1336
1338

FIG. 10

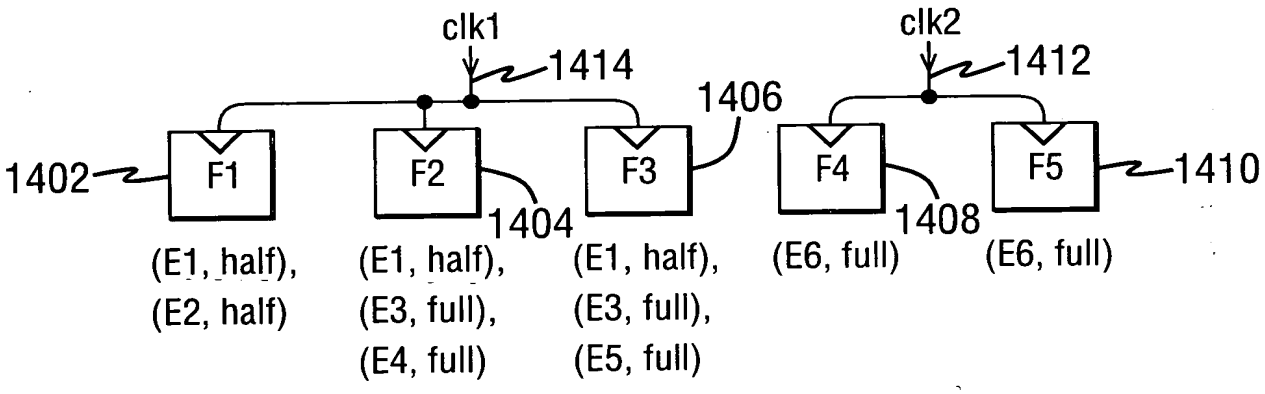

clk1
1414
clk2
1412
1406
1402
F1
F2
F3
F4
F5
1410
1404
1408
(E1, half),
(E2, half)
(E1, half),
(E3, full),
(E4, full)
(E1, half),
(E3, full),
(E5, full)
(E6, full)
(E6, full)

FIG. 11

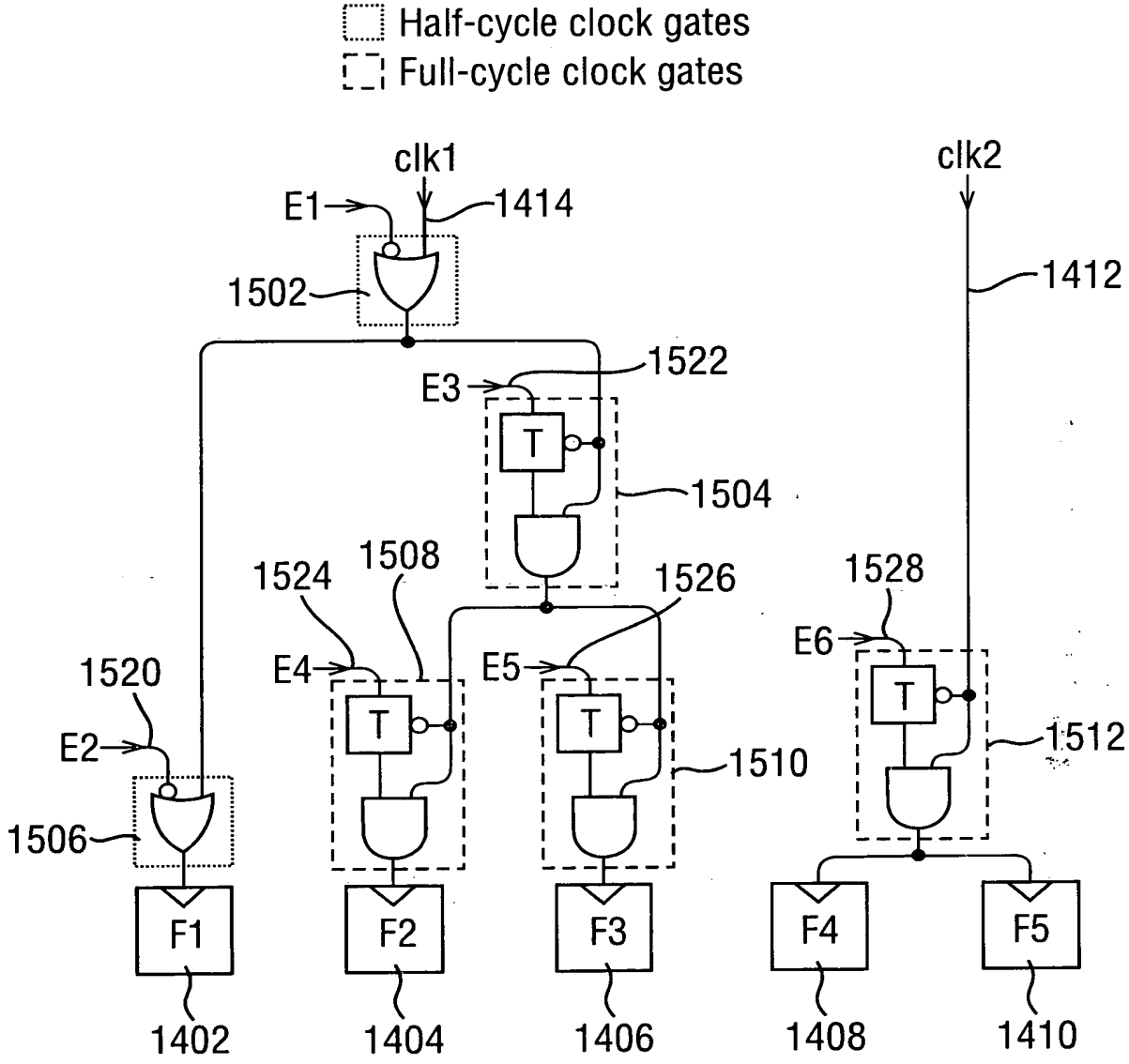

Half-cycle clock gates
Full-cycle clock gates
clk1
clk2
E1
1414
1502
1412
E3
1522
T
1504
1524
1508
1526
1528
E4
E5
E6
1520
E2
1506
1510
1512
F1
F2
F3
F4
F5
1402
1404
1406
1408
1410

FIG. 12

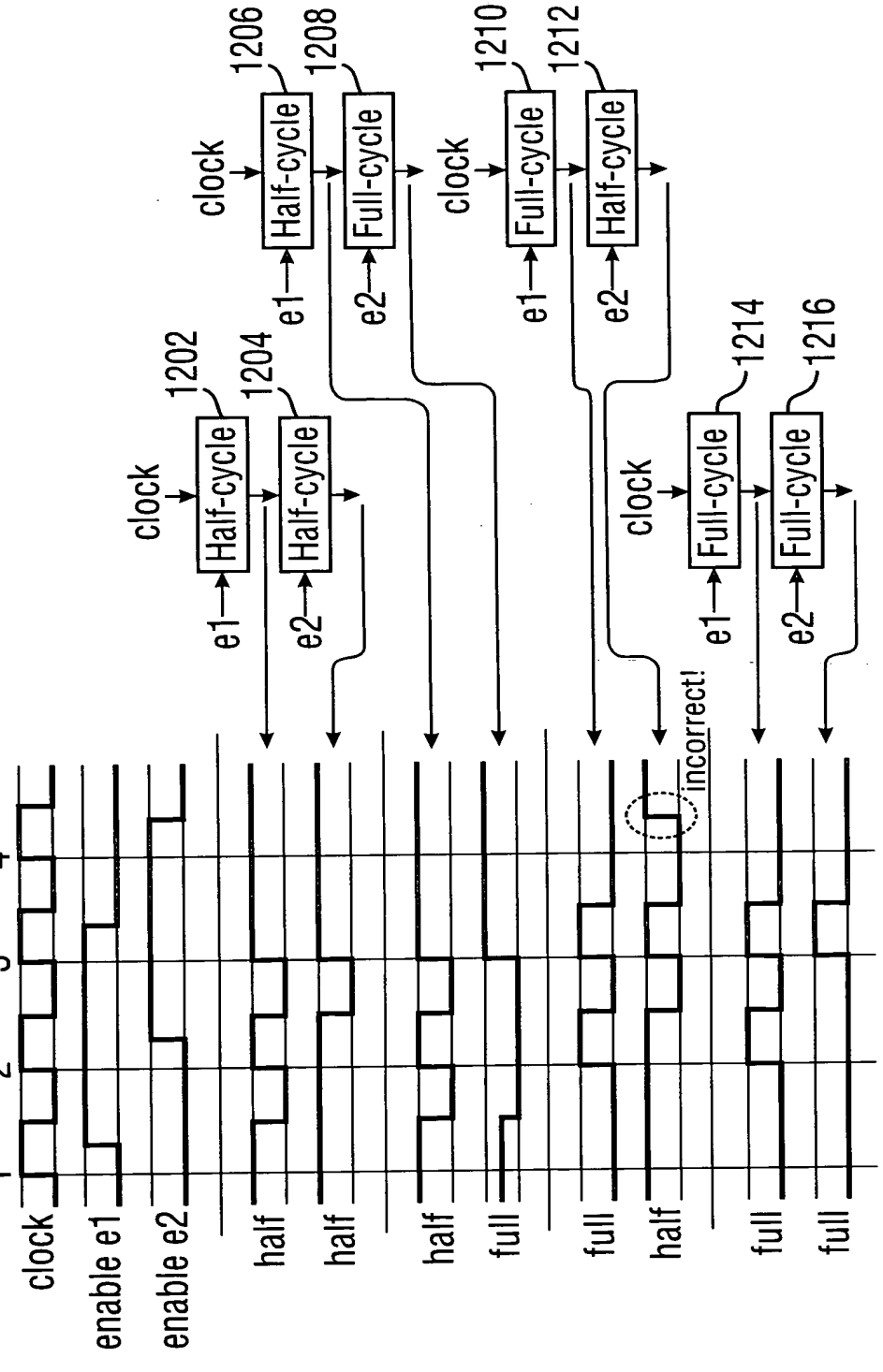

1
2
3
4
clock
enable e1
enable e2
half
half
half
full
full
half
full
full
incorrect!
clock
Half-cycle
1202
e1
clock
Half-cycle
1204
e2
clock
Half-cycle
1206
e1
clock
Full-cycle
1208
e2
clock
Full-cycle
1210
e1
clock
Half-cycle
1212
e2
clock
Full-cycle
1214
e1
clock
Full-cycle
1216
e2

FIG. 13

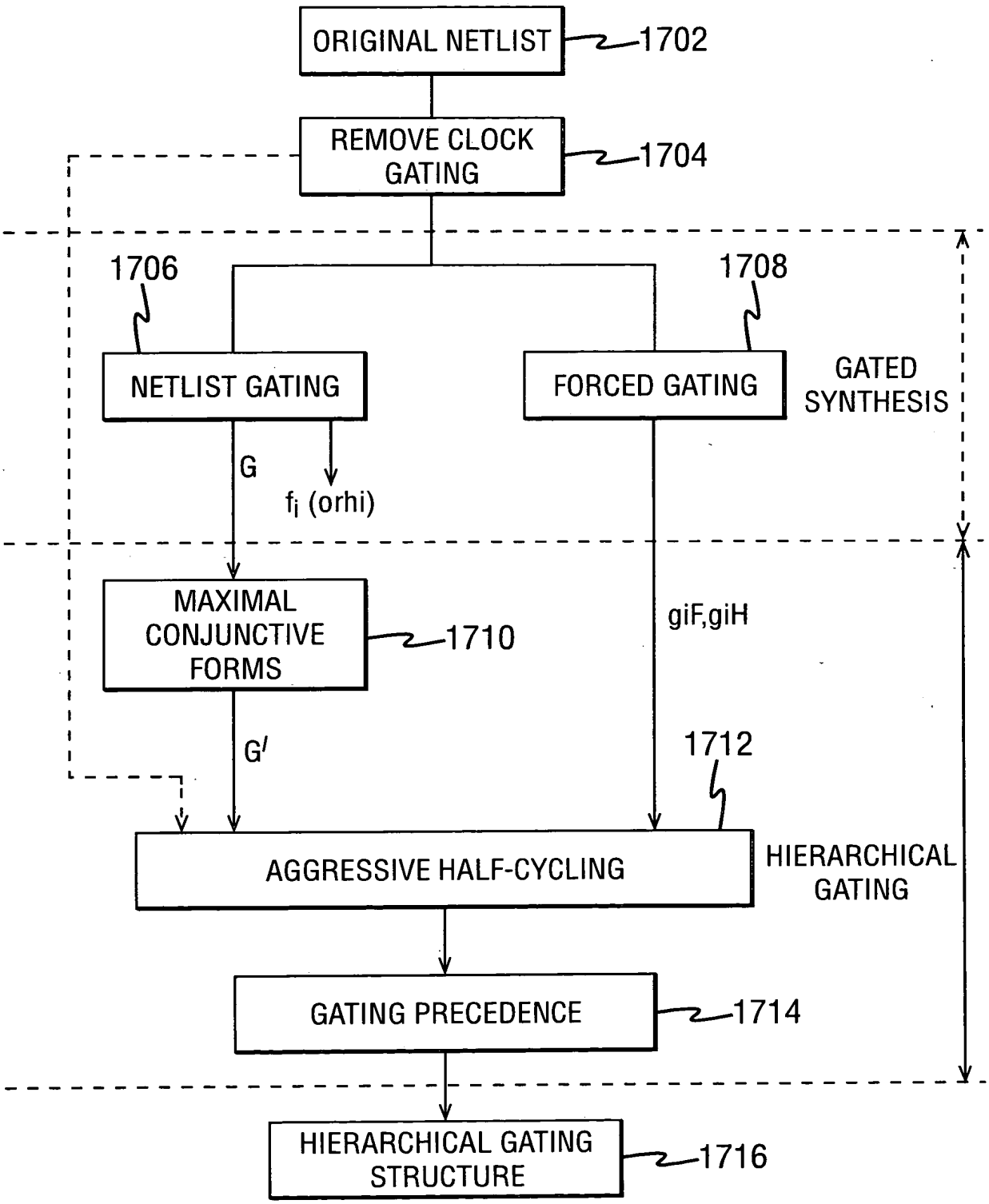

ORIGINAL NETLIST
1702
REMOVE CLOCK GATING
1704
1706
1708
NETLIST GATING
FORCED GATING
GATED SYNTHESIS
G
f_i (orhi)
MAXIMAL CONJUNCTIVE FORMS
1710
giF,giH
G'
1712
AGGRESSIVE HALF-CYCLING
HIERARCHICAL GATING
GATING PRECEDENCE
1714
HIERARCHICAL GATING STRUCTURE
1716

FIG. 14

COMMON SET FINDING

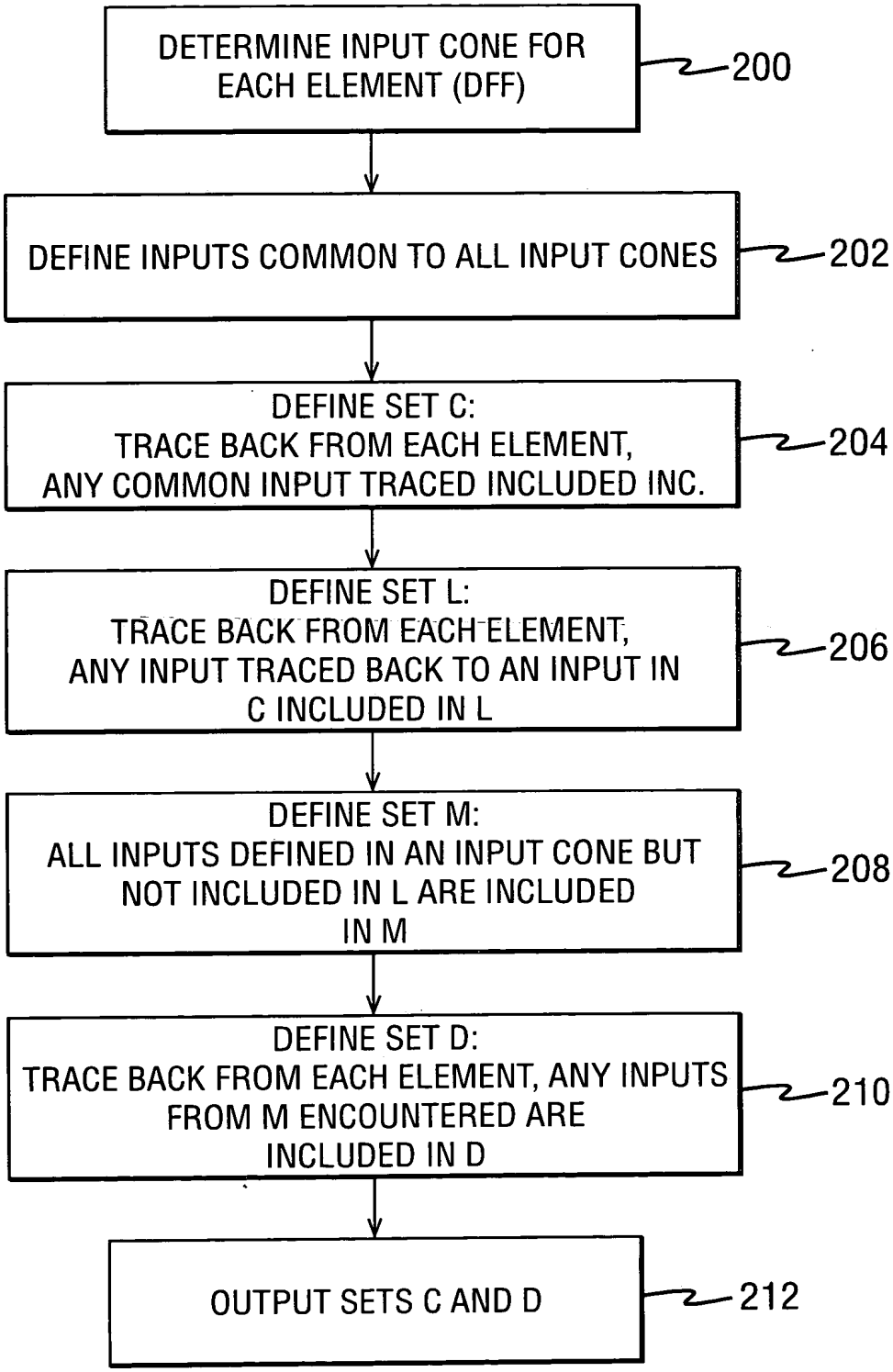


FIG. 15

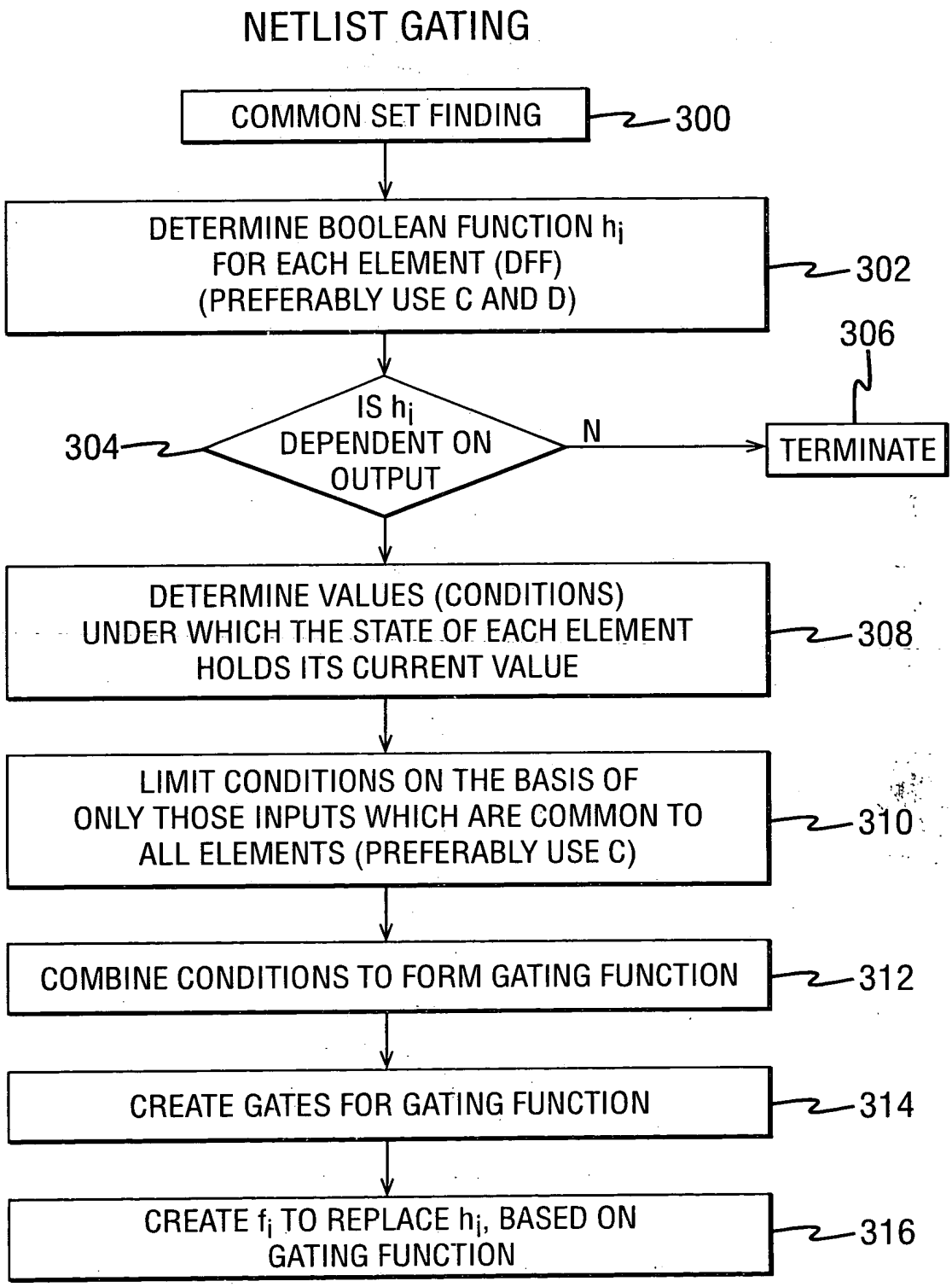

NETLIST GATING
COMMON SET FINDING
300
DETERMINE BOOLEAN FUNCTION h_j FOR EACH ELEMENT (DFF) (PREFERABLY USE C AND D)
302
306
IS h_j DEPENDENT ON OUTPUT
304
N
TERMINATE
DETERMINE VALUES (CONDITIONS) UNDER WHICH THE STATE OF EACH ELEMENT HOLDS ITS CURRENT VALUE
308
LIMIT CONDITIONS ON THE BASIS OF ONLY THOSE INPUTS WHICH ARE COMMON TO ALL ELEMENTS (PREFERABLY USE C)
310
COMBINE CONDITIONS TO FORM GATING FUNCTION
312
CREATE GATES FOR GATING FUNCTION
314
CREATE f_i TO REPLACE h_j, BASED ON GATING FUNCTION
316

FIG. 16

FORCED GATING

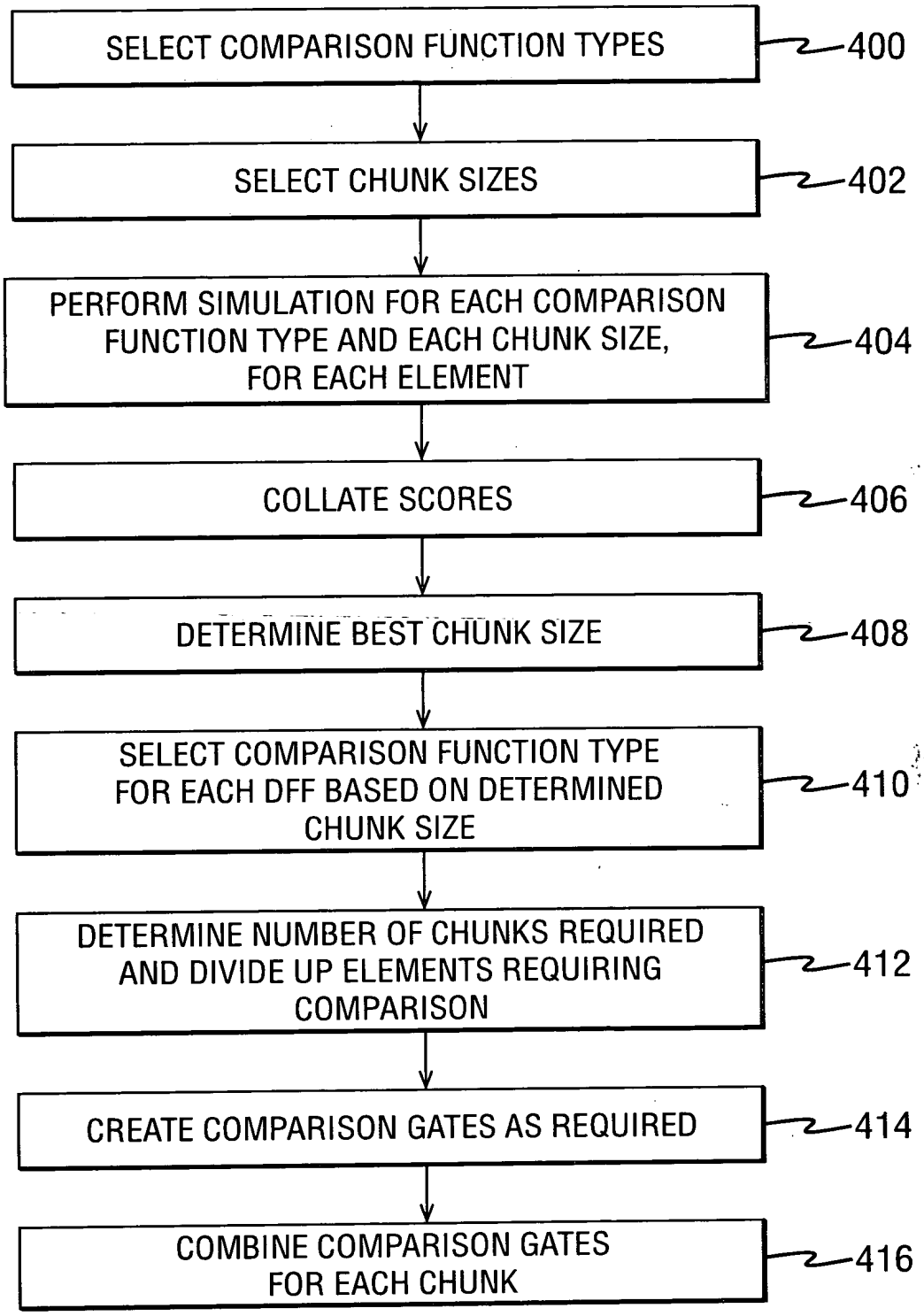
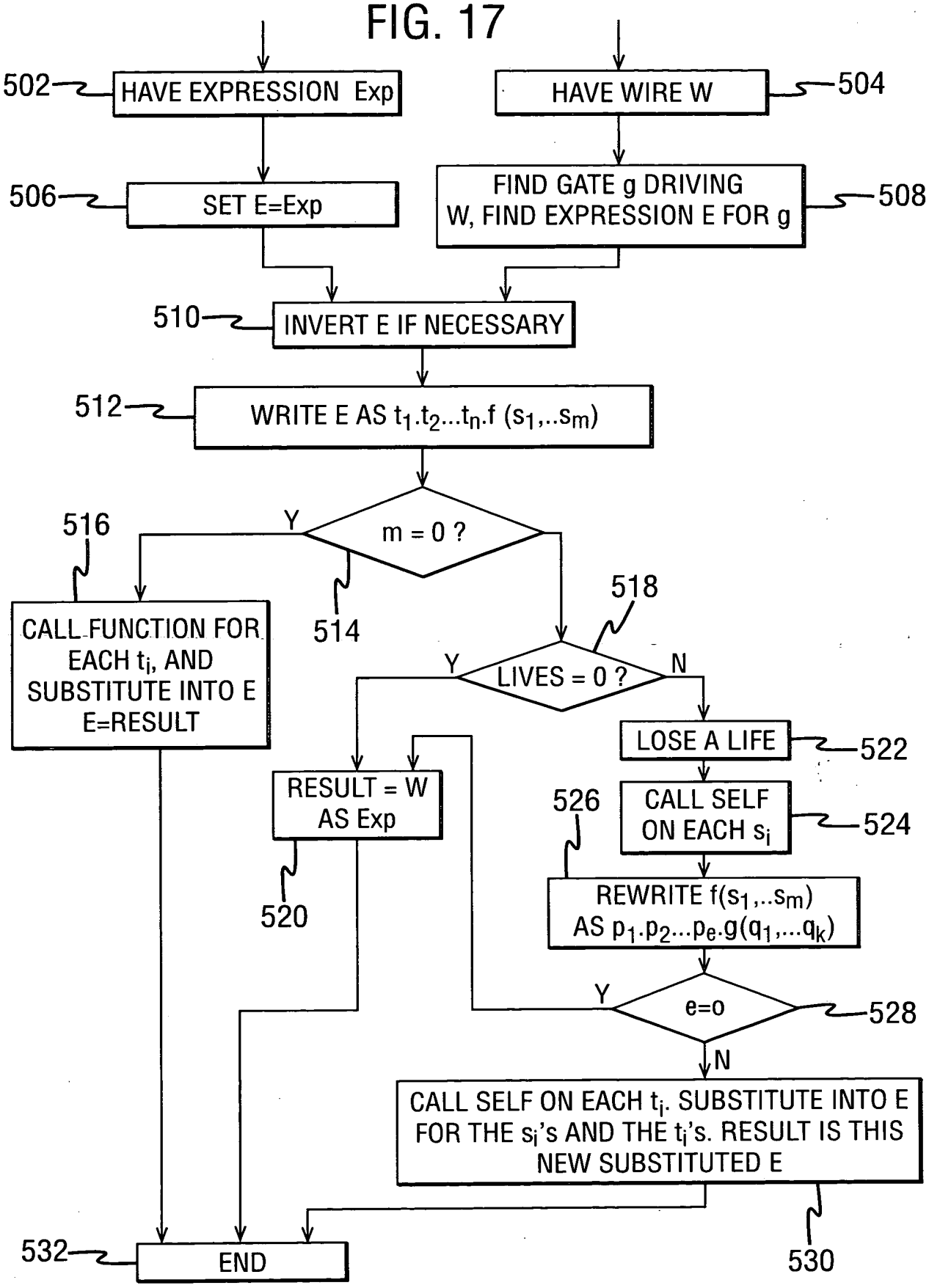


FIG. 17


502
HAVE EXPRESSION Exp
504
HAVE WIRE W
506
SET E=Exp
FIND GATE g DRIVING W, FIND EXPRESSION E FOR g
508
510
INVERT E IF NECESSARY
512
WRITE E AS $t_1.t_2...t_n.f(s_1,..s_m)$
516
Y
m = 0 ?
514
518
CALL FUNCTION FOR EACH t_i, AND SUBSTITUTE INTO E E=RESULT
Y
LIVES = 0 ?
N
LOSE A LIFE
522
RESULT = W AS Exp
526
CALL SELF ON EACH s_i
524
520
REWRITE $f(s_1,..s_m)$ AS $p_1.p_2...p_e.g(q_1,...q_k)$
Y
e=o
528
N
CALL SELF ON EACH t_i. SUBSTITUTE INTO E FOR THE s_i's AND THE t_i's. RESULT IS THIS NEW SUBSTITUTED E
530
532
END